

LAPORAN TUGAS PEMROGRAMAN BERBASIS FRAMEWORK

Modul 9
Global API

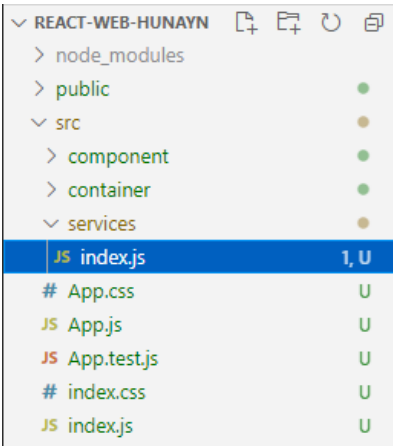
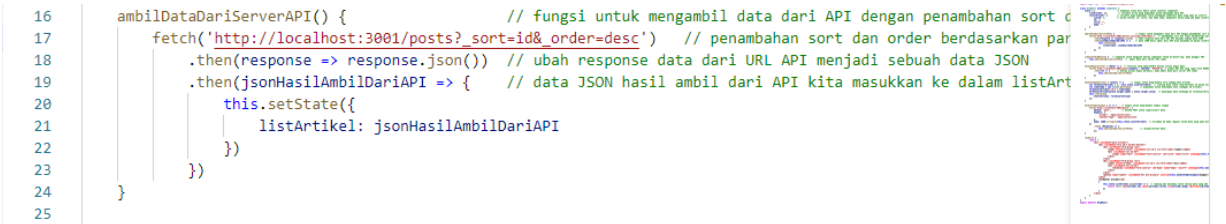


Disusun Oleh :

Nama : Hunayn Risatayn
NIM : 1841720148
Kelas : TI-3E

Program Studi D-IV Teknik Informatika
Jurusan Teknologi Informasi
Politeknik Negeri Malang
April 2021

Praktikum – Bagian 1: Global API Service GET

Langkah	Keterangan
1	<p>Buat folder baru bernama “Services” dalam folder src, kemudian buat file index.js</p>  <p>Buka file BlogPost.jsx pada folder container (statefull component) dan akan tampil kode program seperti Gambar 1.2</p>  <p>Pada fungsi ambilDataDariServerAPI (baris16) terdapat pemanggilan API GET untuk merequest data artikel. Proses dari fungsi inilah yang akan kita manage kedalam satu tempat yaitu index.js.</p>  <p>Buka file index.js pada folder service, yang telah kita buat tadi dan tuliskan baris kode seperti Gambar 1.3 berikut</p>

```

src > services > JS index.js > ...
1  const domainPath = "http://localhost:3001"; // simpan url domain server API pada variabel, sehingga bisa dinamis
2  const GetAPI = (path) => {
3    //path digunakan untuk menunjuk alamat API yang akan di-request
4    const promise = new Promise((resolve, reject) => {
5      fetch(`${domainPath}/${path}`) // alamat url domain + path untuk mengakses full alamat API yang di-request
6        .then((response) => response.json()) // response dari server harus dijadikan json
7        .then(
8          (result) => {
9            resolve(result); // jika success menerima response dari server maka resolve response ke user
10           },
11          (err) => {
12            reject(err); // jika terjadi error dari server (server down, dll)
13          }
14        ); // maka kirim pesan error ke user melalui reject
15    });
16    return promise;
17  };
18
19  const getNewsBlog = () => GetAPI("posts?_sort=id&_order=desc");
20  const API = {
21    // inialisasi function-function yang akan disediakan global API
22    getNewsBlog,
23  };
24
25  export default API;
26

```

Kembali ke file BlogPost.js. Ganti baris 17 sampai baris 23 menjadi seperti Gambar 1.4

```

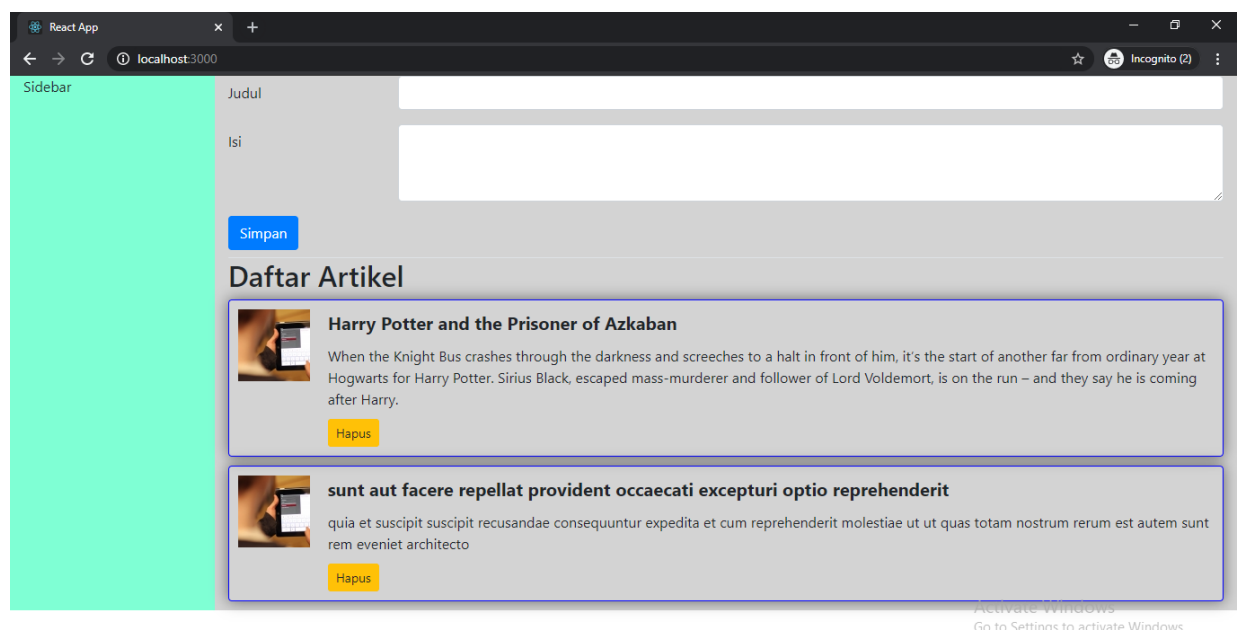
18  ambilDataDariServerAPI() { // fungsi untuk mengambil data dari API dengan penambahan sort dan or
19    API.getNewsBlog().then((result) => {
20      this.setState({
21        listArtikel: result,
22      });
23    });
24  }
25

```

Simpan file BlogPost.js dan index.js tersebut, dan amati apa yang terjadi pada browser kalian.

API is not defined karena kita belum melakukan import class API dari file index.js

Setelah dilakukan import maka akan work



Pertanyaan Praktikum 1

a. Perhatikan file index.js, apa tujuan dibuatnya fungsi GetAPI pada baris 2 dan fungsi getNewsBlog pada baris 16?

Const GetAPI berfungsi untuk melakukan fetch data pada fake Server yaitu localhost:3001

Const getNewsBlog berfungsi sebagai endpoint dari domainPath yang akan melakukan sorting

berdasarkan id kemudian melakukan order secara descending

Global API service POST

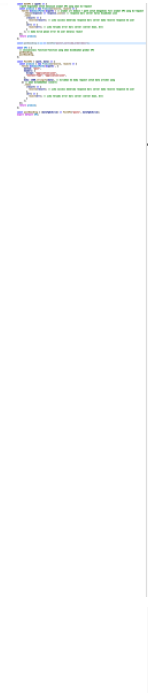
Kita perhatikan pada fungsi handleTombolSimpan pada file BlogPost.js. Bagian inilah yang selanjutnya kita kelola agar bisa di-manage.

```
53 handleTombolSimpan = () => {
54   // fungsi untuk meng-handle tombol simpan
55   fetch("http://localhost:3001/posts", {
56     method: "post", // method POST untuk input/insert data
57     headers: {
58       Accept: "application/json",
59       "Content-Type": "application/json",
60     },
61     body: JSON.stringify(this.state.insertArtikel), // kirimkan ke body request untuk data yang akan ditambahk
62   }).then((Response) => {
63     this.ambilDataDariServerAPI(); // reload/refresh data
64   });
65 };
66
```



Buatlah fungsi untuk menampung action POST dari file BlogPost.js pada file services/index.js dan inialisasi fungsi tersebut seperti pada Gambar 2.2

```
22 const API = {
23   // inialisasi function-function yang akan disediakan global API
24   getNewsBlog,
25   postNewsBlog,
26 };
27
28 const PostAPI = (path, data) => {
29   const promise = new Promise((resolve, reject) => {
30     fetch(`${domainPath}/${path}`, {
31       method: "post",
32       headers: {
33         Accept: "application/json",
34         "Content-Type": "application/json",
35       },
36       body: JSON.stringify(data), // kirimkan ke body request untuk data artikel yang
37     }) // akan ditambahkan (insert)
38     .then(
39       (result) => {
40         resolve(result); // jika success menerima response dari server maka resolve response ke user
41       },
42       (err) => {
43         reject(err); // jika terjadi error dari server (server down, dll)
44       }
45     );
46   });
47   return promise;
48 };
49
50 const postNewsBlog = (dataYgDiKirim) => PostAPI("posts", dataYgDiKirim);
51 export default API;
52
```



Selanjutnya pindah ke file BlogPost.js dan ganti isi dari fungsi handleTombolSimpan menjadi seperti Gambar 2.3

```
53 handleTombolSimpan = () => {
54   // fungsi untuk meng-handle tombol simpan
55   API.postNewsBlog(this.state.insertArtikel).then((response) => {
56     this.ambilDataDariServerAPI();
57   });
58 };
59
```

Selesai. Kode program pada BlogPost.js sedikit lebih simple dari seelumnya

Silahkan kalian jalankan proses input artikel melalui browser dan amati apa yang terjadi

Data akan bertambah ke file listArtikel.json sehingga hasil fetch data dari server localhost:3001 akan mengikuti data yang telah diupdate pada listArtikel.json

Pertanyaan Praktikum 2

1. Perhatikan file index.js, apa tujuan dibuatnya fungsi PostAPI dan fungsi postNewsBlog?
Const postAPI digunakan untuk melakukan post data atau insert data ke server API localhost:3001 ditandai dengan adanya method: post atau berperan sebagai model pada struktur MVC
2. Pada fungsi postNewsBlog, terdapat variable dataYangDiKirim. Apa tujuan dari dibuatnya variable tersebut?
dataYgDiKirim berfungsi untuk menampung data sementara dari input user pada state insertArtikel

Global API service DELETE

Kita perhatikan pada fungsi handleHapusArtikel pada file BlogPost.jsx. Bagian inilah yang selanjutnya kita kelola agar bisa di-manage.

```
33 handleHapusArtikel = (data) => {  
34   // function yang meng-handle button action hapus data  
35   fetch(`http://localhost:3001/posts/${data}`, { method: "DELETE" }) // alamat URL API yang ingin kita HAPUS d  
36   .then((res) => {  
37     // ketika proses hapus berhasil, maka ambil data dari server API lokal  
38     this.ambilDataDariServerAPI();  
39   });  
40 };  
41
```



Buatlah fungsi untuk menampung action DELETE dari file BlogPost.jsx pada file services/index.js dan inialisasi fungsi tersebut seperti pada Gambar 3.2

```
20 const getNewsBlog = () => GetAPI("posts?_sort=id&_order=desc");  
21 const postNewsBlog = (dataYgDiKirim) => PostAPI("posts", dataYgDiKirim);  
22 const deleteNewsBlog = (dataYgDiHapus) => DeleteAPI("posts", dataYgDiHapus);  
23  
24 const API = {  
25   // inialisasi function-function yang akan disediakan global API  
26   getNewsBlog,  
27   postNewsBlog,  
28   deleteNewsBlog,  
29 };  
30
```



```
53 const DeleteAPI = (path, data) => {  
54   const promise = new Promise((resolve, reject) => {  
55     fetch(`${domainPath}/${path}/${data}`, { method: "DELETE" }) // alamat URL API yang ingin kta hapus datanya  
56     .then(  
57       (result) => {  
58         resolve(result); // jika success menerima response dari server maka resolve response ke user  
59       },  
60       (err) => {  
61         reject(err); // jika terjadi error dari server (server down, dll)  
62       }  
63     );  
64   });  
65 };  
66
```



Selanjutnya pindah ke file BlogPost.jsx dan ganti isi dari fungsi handleHapusArtikel menjadi seperti Gambar 3.3

```
33 handleHapusArtikel = (data) => {  
34   // function yang meng-handle button action hapus data  
35   API.deleteNewsBlog(data);  
36   this.ambilDataDariServerAPI();  
37 };  
38
```

Selesai. Kode program pada BlogPost.jsx sedikit lebih simple dari seelumnya.

Silahkan kalian jalankan proses hapus artikel melalui browser dan amati apa yang terjadi

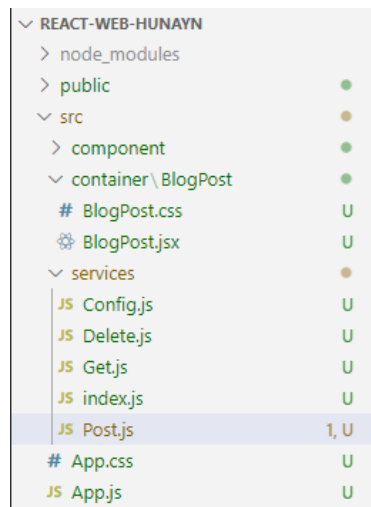
Data dengan id yang dipilih akan terhapus dari file listArtikel.json apabila user mengklik button delete pada web

Pertanyaan Praktikum 3

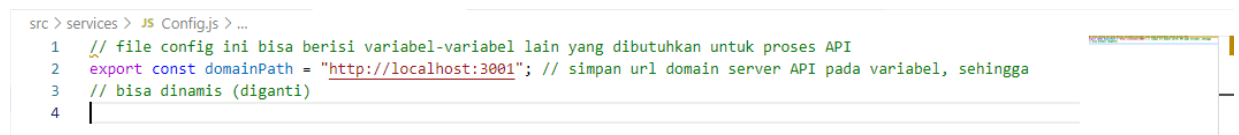
1. Perhatikan file index.js, apa tujuan dibuatnya fungsi DeleteAPI dan fungsi deleteNewsBlog?
Const deleteAPI digunakan untuk melakukan method: 'DELETE' pada server API localhost:3001
Const deleteNewsBlog berfungsi untuk melakukan mapping dengan mengirimkan dataYgDihapus berupa idArtikel ke const DeleteAPI
2. Pada fungsi deleteNewsBlog, terdapat variable dataYangDiHapus. Apa tujuan dari dibuatnya variable tersebut?
Untuk menampung data berupa idArtikel yang dihapus pada BlogPost.jsx

Manage Global API service

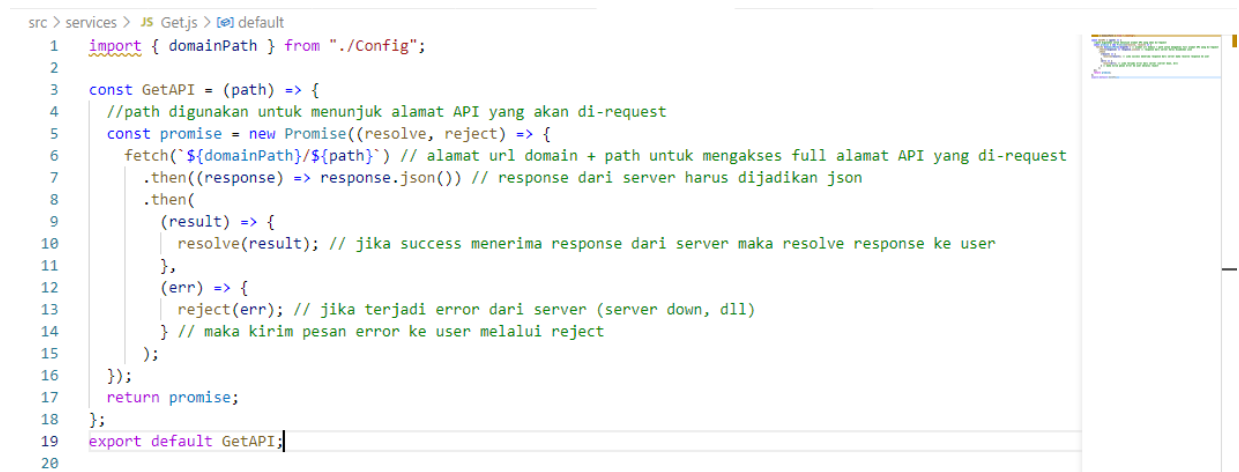
Buatlah file Config.js, Get.js, Post.js, dan Delete.js dalam folder services seperti pada Gambar 4.1



Buka Config.js dan isikan kode seperti Gambar 4.2



Buka Get.js dan isikan kode seperti Gambar 4.3



Buka Post.js dan isikan kode seperti Gambar 4.4

```

src > services > JS Postjs > default
1  import { domainPath } from "../Config";
2
3  const PostAPI = (path, data) => {
4    const promise = new Promise((resolve, reject) => {
5      fetch(`${domainPath}/${path}`, {
6        method: "post", // method post untuk insert data
7        headers: {
8          Accept: "application/json",
9          "Content-Type": "application/json",
10         },
11        body: JSON.stringify(data), // kirimkan ke body request untuk data artikel yang
12      }) // akan ditambahkan (insert)
13    ).then(
14      (result) => {
15        resolve(result); // jika success menerima response dari server maka resolve response ke user
16      },
17      (err) => {
18        reject(err); // jika terjadi error dari server (server down, dll)
19      }
20    );
21  });
22  return promise;
23 };
24 export default PostAPI;
25

```

Buka Delete.js dan isikan kode seperti Gambar 4.5

```

src > services > JS Deletejs > ...
1  import { domainPath } from "../Config";
2
3  const DeleteAPI = (path, data) => {
4    const promise = new Promise((resolve, reject) => {
5      fetch(`${domainPath}/${path}/${data}`, { method: "DELETE" }) // alamat URL API yang ingin kita hapus datanya
6    ).then(
7      (result) => {
8        resolve(result); // jika success menerima response dari server maka resolve response ke user
9      },
10     (err) => {
11       reject(err); // jika terjadi error dari server (server down, dll)
12     }
13   );
14 });
15 };
16 export default DeleteAPI;
17

```

Pada index.js kita edit menjadi seperti Gambar 4.6

```

src > services > JS index.js > ...
1  import GetAPI from "../Get";
2  import PostAPI from "../Post";
3  import DeleteAPI from "../Delete";
4
5  // Daftar API - GET
6  const getNewsBlog = () => GetAPI("posts?_sort=id&_order=desc");
7  // Daftar API - POST
8  const postNewsBlog = (dataYgDiKirim) => PostAPI("posts", dataYgDiKirim);
9  // Daftar API - DELETE
10 const deleteNewsBlog = (dataYgDiHapus) => DeleteAPI("posts", dataYgDiHapus);
11
12 const API = {
13   // inialisasi function-function yang akan disediakan global API
14   getNewsBlog,
15   postNewsBlog,
16   deleteNewsBlog,
17 };
18
19 export default API;
20

```

Amati apa yang terjadi

Project berjalan seperti praktikum sebelumnya dengan manajemen function yang lebih teratur dan lebih mudah dilakukan troubleshooting

Pertanyaan Praktikum 4

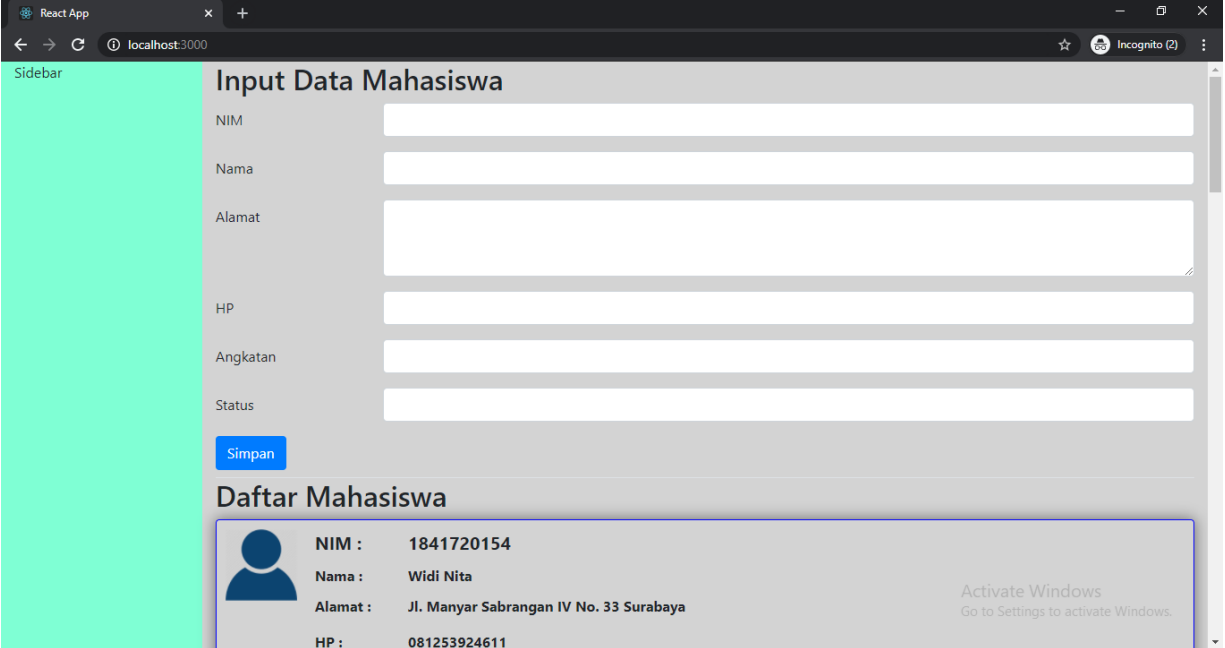
1. Bagaimana caranya untuk menambahkan daftar API baru baik untuk method GET, POST, DELETE pada

Global API?

Dilakukan dengan cara memanggil const yang telah diimport dari file atau kelas lain dan melakukan passing parameter ketika dibutuhkan seperti pada method POST dan DELETE

TUGAS PRAKTIKUM

Ubahlah Tugas pada Modul 4 yang sudah kalian buat dengan memanfaatkan Global API seperti praktikum yang kita lakukan pada Modul 8 ini.



The screenshot shows a web application running on localhost:3000. The application has a sidebar on the left and a main content area. The main content area is divided into two sections: "Input Data Mahasiswa" and "Daftar Mahasiswa".

The "Input Data Mahasiswa" section contains form fields for NIM, Nama, Alamat, HP, Angkatan, and Status, along with a "Simpan" button.

The "Daftar Mahasiswa" section displays a list of student data, including a profile icon, NIM, Nama, Alamat, and HP.

NIM	Nama	Alamat	HP
1841720154	Widi Nita	Jl. Manyar Sabrangan IV No. 33 Surabaya	081253924611

Link GitHub : https://github.com/hunaynr/framework_prog/tree/main/9thWeek

Link Youtube : <https://youtu.be/gzYnYug6RFs>

